

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS - *CAMPUS* BETIM
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Gabriel Vila Nova de Paula

**SISTEMA AUTOMATIZADO DE IRRIGAÇÃO PARA CULTIVO RESIDENCIAL DE
TOMATES**

Betim
2025

GABRIEL VILA NOVA DE PAULA

**SISTEMA AUTOMATIZADO DE IRRIGAÇÃO PARA CULTIVO RESIDENCIAL DE
TOMATES**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Arthur Hermano Rezende Rosa

Betim

2025

FICHA CATALOGRÁFICA

P324s Paula, Gabriel Vila Nova de
Sistema automatizado de irrigação para cultivo residencial de
tomates / Gabriel Vila Nova de Paula. – 2025.

72 f.: il.

Trabalho de conclusão de curso (Bacharelado em
Engenharia de Controle e Automação) - Instituto Federal de Educação,
Ciência e Tecnologia de Minas Gerais, Campus Betim, 2025.

Orientação: Prof. Dr. Arthur Hermano Rezende Rosa

1. Irrigação. 2. Prototipagem. 3. Microcontroladores. 4. Arduino. 5.
Engenharia de Controle e Automação. I. Paula, Gabriel Vila Nova de. II.
Título.

CDU: 681.51



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Betim
Diretoria de Ensino
Docentes Automação Industrial e Tecnologia da Informação
Rua Itamarati - CEP 32677-564 - Betim - MG
3135976360 - www.ifmg.edu.br

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Aos seis dias do mês de fevereiro do ano de 2025, às dezoito horas, nas dependências do IFMG – *Campus* Betim reuniu-se a banca examinadora presidida por mim, Arthur Hermano Rezende Rosa e demais membros, Daniel Godinho e Mauricio Monteiro da Silva. Nesta ocasião o discente GABRIEL VILA NOVA DE PAULA do curso de Bacharelado em Engenharia de Controle e Automação defendeu seu Trabalho de Conclusão de Curso (TCC) intitulado “SISTEMA AUTOMATIZADO DE IRRIGAÇÃO PARA CULTIVO RESIDENCIAL DE TOMATES” e foi APROVADO (com ressalvas) com 65 pontos.

O lançamento da nota e o conseqüente encerramento do respectivo processo está condicionado ao cumprimento dos procedimentos pós-defesa conforme previstos nos regulamentos vigentes. Tais procedimentos pós-defesa devem ser finalizados dentro do prazo limite de 20 dias, a contar da data desta ata. O descumprimento destes procedimentos até a data estipulada implicará em atribuição de nota 0 (zero) e conseqüente reprovação.

A sessão foi encerrada às 19 horas. Para constar, eu, Arthur Hermano Rezende Rosa, redigi a presente ata que após lida publicamente, foi aprovada e assinada pelo discente e membros da banca examinadora.

Betim, 06 de fevereiro de 2025.



Documento assinado eletronicamente por **Arthur Hermano Rezende Rosa, Professor**, em 10/02/2025, às 11:45, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Mauricio Monteiro da Silva, Professor**, em 11/02/2025, às 16:28, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Daniel Almeida Godinho, Professor Substituto**, em 11/02/2025, às 19:14, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2194283** e o código CRC **CD1095DC**.

AGRADECIMENTOS

Um agradecimento especial aos meus professores do Instituto Federal de Minas Gerais (IFMG). A orientação e apoio de vocês enriqueceram minha jornada de aprendizado e me inspiraram a perseguir meus objetivos.

Agradeço também ao meu orientador, Professor Arthur Hermano. Sua orientação, paciência e dedicação foram importantes para a realização deste trabalho.

Aos meus colegas de turma, obrigado por compartilharem esta jornada comigo, em especial meu profundo agradecimento vai também ao meu amigo Frederico Marazzi, seu apoio constante e incentivo foram essenciais ao longo desta jornada.

E, por último, mas não menos importante, agradeço à minha família. Seu apoio e encorajamento foram a minha maior força. Sem vocês, eu não estaria onde estou hoje. A todos vocês, meu sincero agradecimento. Este trabalho é um reflexo da contribuição e fé que cada um de vocês depositou em mim.

RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo de sistema de irrigação automatizado para vasos de plantas residenciais, com foco no cultivo de tomates. O projeto utiliza microcontroladores Arduino e ESP32, combinados com sensores de umidade, bomba d'água e relés, para automatizar a irrigação com base na umidade do solo. O sistema garante que a irrigação ocorra apenas quando necessário, proporcionando maior praticidade no cuidado com as plantas. Além disso, a proposta enfatiza a utilização de componentes acessíveis, tornando a implementação viável para aplicações domésticas.

Palavras-chave: Automação; Irrigação; Tomates; Prototipagem; ESP32.

ABSTRACT

Home This work presents the development of a prototype for an automated irrigation system designed for residential potted plants, focusing on tomato cultivation. The project utilizes Arduino and ESP32 microcontrollers, combined with soil moisture sensors, a water pump, and relays, to automate irrigation based on soil moisture levels. The system ensures that irrigation occurs only when necessary, providing greater convenience in plant care. Additionally, the proposal emphasizes the use of accessible components, making the implementation feasible for home applications.

Keywords: Automation; Irrigation; Tomato; Prototyping; ESP32.

LISTA DE ILUSTRAÇÕES

Figura 1: Microcontrolador ATmega328P.....	14
Figura 2: Arduino Uno Rev3.....	14
Figura 3: Módulo sensor de umidade de solo.....	15
Figura 4: Sensor de umidade do solo.....	16
Figura 5: Sensor de Nivel de Água 90°.....	18
Figura 6: Módulo Relé 1 Canal 5v.....	20
Figura 7: Minibomba de Água para Arduino.....	21
Figura 8: Mini Bomba Submersível p/ Água.....	22
Figura 9: Logo Arduino.....	25
Figura 10: Ligação 5v e GND Arduino.....	29
Figura 11: Ligação relé no Arduino.....	30
Figura 12: Ligação Sensor de Umidade.....	30
Figura 13: Ligação Sensor de Nivel de Água.....	31
Figura 14: Ligação dos LEDs.....	31
Figura 15: Ligação do Buzzer.....	32
Figura 16: Diagrama do Circuito Elétrico no Tinkercad.....	33
Figura 17: Base de Acrílico com Componentes.....	34
Figura 18: Organização dos Cabos.....	34
Figura 19: Reservatório de Água.....	35
Figura 20: ESP32.....	38
Figura 21: Sensor ultrassónico.....	38
Figura 22: HidroBot Telegram.....	39
Figura 23: Estrutura - Visão lateral.....	41
Figura 24: Estrutura - Visão de cima.....	41
Figura 25: Estrutura - Saida dos cabos de alimentação.....	42
Figura 26: Primeiro teste do sistema.....	43
Figura 27: Exemplos de mensagens Telegram.....	45
Figura 28:Mini Painel Solar Fotovoltaico 5V/1W.....	49

LISTA DE TABELAS

Tabela 1: Especificações Arduino Uno R3 SMD CH340	15
Tabela 2: Especificações Módulo Sensor De Umidade De Solo	17
Tabela 3: Pinagem Módulo Sensor De Umidade De Solo.....	17
Tabela 4: Especificações Sensor de Nível de Água 90°	18
Tabela 5: Especificações Módulo Relé 1 Canal 5v	20
Tabela 6: Especificações Mini Bomba Submersível p/ Água.....	22
Tabela 7: Especificação Buzzer ativo 5V	24
Tabela 8: Custo componentes.....	48

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Justificativa.....	10
1.2	Objetivos	11
1.2.1	Objetivos específicos.....	11
1.3	Estrutura do Trabalho	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Internet das Coisas (IoT).....	12
2.2	Irrigação por Superfície	13
2.3	Microcontrolador	13
2.3.1	Arduino Uno Rev3	14
2.4	Sensores	15
2.4.1	Módulo Sensor De Umidade De Solo.....	16
2.4.2	Sensor de Nível.....	17
2.5	Relé 5V	18
2.5.1	Módulo de relé 5V	19
2.6	Bomba de água.....	20
2.5.1	Mini Bomba Submersível.....	21
2.7	Buzzer.....	23
2.7.1	Buzzer Ativo 5v.....	23
2.8	Arduino IDE.....	24
2.8.1	Download do Arduino IDE.	25
3	METODOLOGIA	26
3.1	Visão geral do projeto	26
3.1.1	Lista dos principais componentes utilizados.....	27
3.2	Estrutura do projeto	27
3.3	Montagem do protótipo	28
3.3.1	Ligações eletrônicas.....	29
3.3.2	Diagrama do Circuito Elétrico	32
3.3.3	Estrutura mecânica.....	33
3.3.4	Programação do sistema - Arduino IDE	35
4	MELHORIAS	37
4.1	ESP32	37
4.2	Sensor ultrassônico	38

4.3	Telegram	39
4.3.1	Chatbot.....	39
4.4	Programação do novo sistema.	40
4.5	Estrutura mecânica	40
5	RESULTADOS.....	42
5.1	Funcionamento do Sistema com Arduino.....	42
5.1.1	Desempenho do Sensor de Umidade.....	43
5.1.2	Funcionamento do Sensor de Nível de Água	44
5.1.3	Atuação do Relé e Bomba de Água	44
5.1.4	Indicadores Visuais e Sonoros	44
5.2	Funcionamento do Sistema após melhorias	45
5.2.1	ESP32 e Monitoramento Remoto	45
5.2.2	Desempenho do Sensor Ultrassônico	47
5.2.3	Indicadores Visuais e Sonoros	48
5.3	Custos de Implementação do Sistema de Irrigação.....	48
5.4	Possibilidades de outras melhorias	49
6	CONCLUSÃO	49
	REFERÊNCIAS.....	51
	APÊNDICE A	53
	APÊNDICE B	58

1 INTRODUÇÃO

Este trabalho busca explorar o potencial dos sistemas automáticos de irrigação, utilizando como referência o cultivo residencial de tomates. Serão abordados os aspectos técnicos relacionados à implementação de um protótipo viável para uso em ambientes residenciais. A partir da análise de soluções existentes e da proposição de melhorias no campo da irrigação inteligente, pretende-se contribuir para o avanço de práticas de jardinagem mais acessíveis, atendendo às necessidades dos ambientes urbanos.

Os tomates estão entre os vegetais mais cultivados em jardins domésticos, devido à sua relativa facilidade de cultivo. No entanto, seu desenvolvimento saudável requer cuidados específicos, sendo a irrigação um dos fatores mais críticos. A falta de irrigação adequada pode levar ao estresse hídrico, comprometendo o crescimento e a vitalidade das plantas (DA CUNHA FERREIRA, 2021). Para garantir que as plantas de tomate recebam água suficiente para suas funções fisiológicas essenciais, a umidade relativa ideal para cultivos em estufas varia entre 50% e 70% (ALVARENGA, 2022).

Neste contexto, os sistemas de irrigação automatizados podem ser aplicados em residências para o cuidado de hortaliças, plantas e jardins, reduzindo a necessidade de intervenção manual. Ao combinar a técnica de irrigação localizada com um sistema automatizado baseado em sensores de umidade do solo e outros parâmetros ambientais, é possível desenvolver uma solução acessível e eficiente para o uso residencial. Dessa forma, a manutenção das plantas se torna mais simples e eficaz, contribuindo para a popularização dessas tecnologias (MEDEIROS, 2018).

1.1 Justificativa

Este estudo justifica-se pela necessidade de integrar tecnologias de automação à jardinagem residencial, com foco no cultivo de tomates. A implementação de um sistema automatizado de irrigação, utilizando sensores de umidade do solo e sensores de nível de água, possibilita uma solução eficiente e prática. Além de eliminar a necessidade de monitoramento manual constante, o sistema permite que os níveis de umidade sejam acompanhados remotamente, oferecendo maior conveniência ao usuário.

1.2 Objetivos

Este trabalho tem como objetivo geral desenvolver um sistema de irrigação automatizado para o cultivo residencial de tomates, ajustando a irrigação com base na umidade do solo, que deve ser mantida entre 50% e 70%. Além disso, busca-se possibilitar o monitoramento remoto do sistema por meio de um chatbot no Telegram, proporcionando maior praticidade e controle ao usuário.

1.2.1 Objetivos específicos

- Implementar sensores de umidade do solo e nível de água para monitorar as condições do sistema em tempo real e acionar a irrigação automaticamente quando necessário.
- Desenvolver a integração do sistema com o Telegram, permitindo que o usuário receba notificações e consulte os níveis de umidade e água remotamente.
- Construir um protótipo funcional utilizando componentes de fácil acesso, garantindo a viabilidade da automação da irrigação para uso residencial.

1.3 Estrutura do Trabalho

Capítulo 1, consiste na descrição geral e as funcionalidades do projeto, delineando a solução proposta e os objetivos detalhados que se pretende alcançar.

Capítulo 2 é uma exploração da base teórica, onde são explorados os conceitos fundamentais e introduzidos os componentes que serão empregados no desenvolvimento do projeto.

Capítulo 3 se dedica à apresentação das ferramentas selecionadas para a execução do projeto e à descrição detalhada de cada etapa do processo de construção.

Capítulo 4 - Este capítulo discute as melhorias para o sistema, incluindo a integração com o ESP32, a adição do sensor ultrassônico e a integração com a plataforma Telegram.

Capítulo 5, apresenta os resultados alcançados e examinados seguindo os objetivos da proposta inicial, oferecendo uma análise crítica do que foi observado.

Capítulo 6 traz as conclusões principais do estudo, destacando as contribuições significativas e sugerindo direções para investigações futuras no campo de estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Este projeto tem como objetivo o desenvolvimento de um modelo experimental, utilizando abordagens de engenharia e instrumentação específicas. Para isso, será apresentada uma fundamentação teórica que abordará os principais componentes e conceitos envolvidos, proporcionando uma compreensão mais detalhada de sua aplicação no sistema proposto.

2.1 Internet das Coisas (IoT)

O termo Internet das Coisas surgiu em 1999, quando o britânico Kevin Ashton propôs um modelo no qual computadores poderiam coletar e processar informações sem intervenção humana, otimizando a produção industrial da época. Com o avanço da tecnologia, essa ideia evoluiu e tornou-se viável, sendo aplicada em diversas áreas, como saúde, transporte, agricultura e automação residencial (SILVA, 2023).

A Internet das Coisas é um paradigma inovador que vem crescendo rapidamente e se consolidando no cenário tecnológico atual, impulsionado pela expansão das redes Wi-Fi e telecomunicações. Esse conceito baseia-se na interconexão de dispositivos inteligentes, como sensores, smartphones e eletrodomésticos, permitindo a comunicação e a troca de informações entre eles de forma autônoma (SILVA, 2023).

2.2 Irrigação por Superfície

No método de irrigação por superfície, a distribuição se dá por gravidade através da superfície do solo. Para isto, é exigida uma condição superficial adequada do solo, de modo a proporcionar um escoamento contínuo sem causar erosão (ALTOÉ, 2012).

Esse método é um dos mais antigos e amplamente utilizados devido à sua simplicidade e baixo custo de implementação. No contexto deste trabalho, embora o sistema utilize uma bomba e mangueiras para conduzir a água até a planta, o método de irrigação adotado se assemelha à irrigação por superfície, pois a água é aplicada diretamente no solo ao redor da planta, garantindo sua absorção gradual e eficiente.

2.3 Microcontrolador

Um microcontrolador é um computador em um chip, embarcado no interior de algum outro dispositivo, contém memória, periféricos de entrada e saída, e um processador, podendo controlar suas funções ou ações (BARBOSA, 2019).

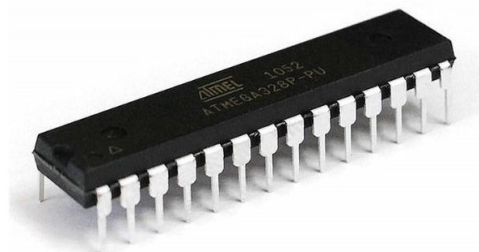
As funções básicas dos microcontroladores para um sistema de irrigação são as seguintes:

- **Programar o sistema:** o microcontrolador deve permitir que o usuário programe o sistema de irrigação de acordo com as necessidades das plantas e as condições climáticas. O usuário deve ser capaz de definir os horários de irrigação, a duração e a frequência da irrigação.
- **Detectar falhas:** o microcontrolador deve ser capaz de detectar problemas no sistema, como falhas nas válvulas ou nos sensores, e alertar o usuário sobre esses problemas.
- **Ser fácil de usar:** o microcontrolador deve ser fácil de instalar e programar, com uma interface amigável para o usuário.

Em resumo, o controlador é um componente fundamental de um sistema de irrigação automatizado residencial e deve ser escolhido com cuidado, levando em

consideração as necessidades do jardim, as condições climáticas e as preferências do usuário.

Figura 1: Microcontrolador ATmega328P.



Fonte: Easytronics, 2023

2.3.1 Arduino Uno Rev3

O Arduino Uno Rev3 é uma placa de microcontrolador baseado no ATmega328. Possui 14 entradas / saídas digitais (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB, um conector de alimentação, um cabeçalho ICSP, e um botão de reset (BARBOSA, 2019).

Em resumo, oferece uma alternativa de baixo custo ao Arduino Uno tradicional, mantendo a flexibilidade e facilidade de uso que tornaram a plataforma Arduino tão popular. Seu uso do chip CH340 para a comunicação USB e o microcontrolador Atmega328 o tornam uma opção econômica sem sacrificar a funcionalidade ou a compatibilidade com o ecossistema Arduino.

Figura 2: Arduino Uno Rev3



Fonte: Eletrogate, 2022

Tabela 1: Especificações Arduino Uno R3 SMD CH340

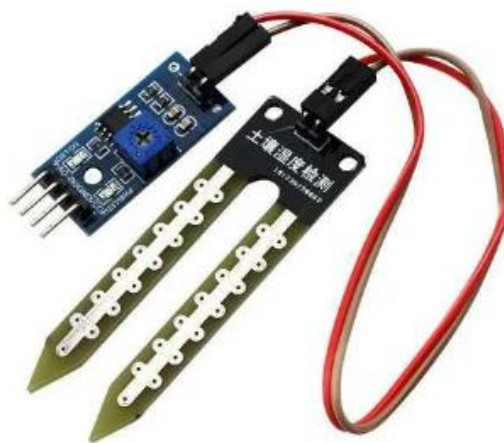
MICROCONTROLADOR	ATmega328
TENSÃO DE OPERAÇÃO	5V
TENSÃO DE ENTRADA	7-9V
PORTAS DIGITAIS	14 (6 podem ser usadas como PWM)
PORTAS ANALÓGICAS	6
CORRENTE PINOS I/O	40 mA (pico/curta duração);
	20 mA (contínua)
CORRENTE PINOS 3,3V	50mA
MEMÓRIA FLASH	32KB (0,5KB usado no bootloader)
SRAM	2KB
EEPROM	1KB
VELOCIDADE DO CLOCK	16MHz

Fonte: Elaborado pelo autor, 2024

2.4 Sensores

Os sensores são componentes essenciais em um sistema de irrigação, pois permitem monitorar as condições do solo, como a umidade e a temperatura, e determinar a quantidade de água necessária para as plantas (VIDAL, 2017).

Figura 3: Módulo sensor de umidade de solo



Fonte: Eletrogate, 2022

2.4.1 Módulo Sensor De Umidade De Solo

O sensor de umidade do solo é um dispositivo essencial para sistemas de irrigação automatizados, permitindo o monitoramento contínuo da umidade do solo e garantindo que a irrigação ocorra apenas quando necessário (VIDAL, 2017).

Seu funcionamento é baseado na medição da resistência elétrica do solo entre duas hastas metálicas condutoras, normalmente feitas de aço inoxidável, que são inseridas no solo. A resistência elétrica varia de acordo com a umidade presente: quanto maior a umidade, menor a resistência elétrica, pois a água facilita a passagem de corrente elétrica. Por outro lado, quando o solo está seco, a resistência aumenta significativamente devido à menor presença de água no meio (VIDAL, 2017).

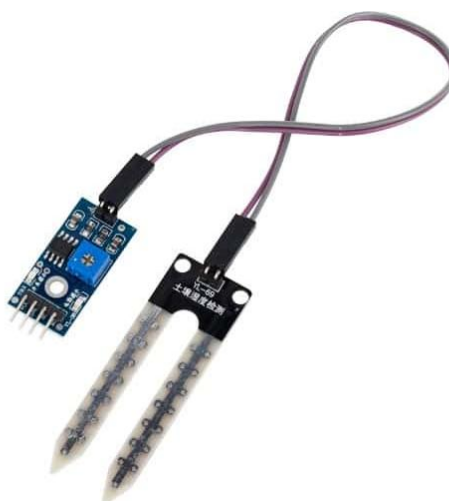
O módulo sensor de umidade utilizado neste projeto possui duas saídas principais:

Saída digital (D0): Fornece um sinal de nível alto ou baixo, dependendo do ajuste de um potenciômetro presente no módulo. Esse ajuste permite definir um limiar de umidade abaixo do qual a irrigação é acionada automaticamente.

Saída analógica (A0): Permite uma leitura mais precisa dos níveis de umidade, fornecendo valores que variam conforme a quantidade de água presente no solo.

Em resumo, o módulo sensor de umidade do solo desempenha um papel fundamental na automação da irrigação, proporcionando um monitoramento eficiente e contribuindo para a conservação dos recursos hídricos.

Figura 4: Sensor de umidade do solo



Fonte: Makerhero, 2023

Tabela 2: Especificações Módulo Sensor De Umidade De Solo

TENSÃO DE OPERAÇÃO	3,3-5V
INFORMAÇÃO	Sensibilidade ajustável via potenciômetro
SAÍDA	Digital e Analógica
LED INDICADOR PARA	tensão (vermelho)
LED INDICADOR PARA	saída digital (verde)
COMPARADOR	LM393
DIMENSÕES PCB	3x1,5 cm
DIMENSÕES Sonda	6x2 cm

Fonte: Elaborado pelo autor, 2024

Tabela 3: Pinagem Módulo Sensor De Umidade De Solo

VCC	3,3-5V
GND	GND
D0	Saída Digital
A0	Saída analógica

Fonte: Elaborado pelo autor, 2024

2.4.2 Sensor de Nível

O sensor de nível de água consiste basicamente em um flutuador conectado a um interruptor mecânico. Quando o nível do líquido muda, o flutuador sobe ou desce junto com o líquido (SOUZA, 2018). Este movimento do flutuador aciona o interruptor, que pode ser configurado para fechar ou abrir um circuito elétrico, enviando um sinal de presença ou ausência de líquido a um controlador ou outro dispositivo de monitoramento (SOUZA, 2018).

Em sistemas de irrigação automatizados, como o proposto para o cultivo residencial de tomates, o Sensor de Nível de Água pode ser integrado ao controlador principal, como um Arduino. Ele pode fornecer feedback em tempo real sobre o nível

de água disponível no reservatório, permitindo ajustes automáticos na programação da irrigação. Se o nível de água estiver abaixo do necessário, o sistema pode interromper a irrigação ou acionar uma bomba para reabastecer o reservatório, garantindo que as plantas recebam a quantidade adequada de água sem desperdícios.

Figura 5: Sensor de Nível de Água 90°



Fonte: Eletrogate, 2022

Tabela 4: Especificações Sensor de Nível de Água 90°

COMPRIMENTO DO CABO	30 (cm)
POTÊNCIA MÁXIMA	10 W
CORRENTE MÁXIMA	0,5 A
TENSÃO MÁXIMA	100 VDC/110 VAC
TEMPERATURA DE OPERAÇÃO	-10 ~ + 85
DIMENSÕES	5,5 cm x 2,5 cm x 3 cm

Fonte: Elaborado pelo autor, 2024

2.5 Relé 5V

O relé de 5V é um componente eletromecânico que atua como um interruptor controlado eletricamente, permitindo o controle de circuitos de alta potência por meio de sinais de baixa tensão. Ao ser energizado com uma corrente contínua de 5V, a bobina interna do relé gera um campo magnético que altera a posição dos contatos,

alternando entre os estados normalmente aberto (NO) e normalmente fechado (NC). Isso possibilita o controle de dispositivos de maior potência, como motores e lâmpadas, utilizando sinais de controle de baixa potência provenientes de microcontroladores como o Arduino ou o ESP32 (WILLIAM, 2022).

As especificações do relé de 5V abrangem:

- Tensão nominal de 5V DC
- Corrente operacional de 70mA
- Capacidade de corrente de carga em CA de até 10A a 250V CA ou 125V CA
- Capacidade de corrente de carga em CC de até 10A a 30V CC ou 28V CC
- Constituído por 5 pinos e fabricado com material plástico
- Tempo de ativação de 10ms e tempo de desativação de 5ms
- Capacidade de até 300 comutações por minuto

2.5.1 Módulo de relé 5V

Este módulo, composto por uma única placa de canal, é projetado para controlar cargas de alta tensão e corrente, como motores, válvulas solenoides, cargas CA e iluminação. Desenvolvido para ser conectado a diversos microcontroladores, como PIC, Arduino, entre outros, o relé permite a ativação ou desativação dos contatos através de uma bobina que, ao ser energizada, cria um campo magnético capaz de unir os contatos. Uma mola os separa novamente quando a energia na bobina é desligada (WILLIAM, 2022).

No contexto de um sistema de irrigação automatizado, o módulo de relé de 5V pode ser utilizado para controlar a ativação de bombas de água ou válvulas solenoides, permitindo o gerenciamento eficiente da irrigação com base nas leituras de sensores de umidade do solo. Isso possibilita a criação de sistemas de irrigação inteligentes que se ajustam automaticamente às necessidades hídricas das plantas e contribuindo para o crescimento saudável do jardim.

Figura 6: Módulo Relé 1 Canal 5v



Fonte: Eletrogate, 2024

Tabela 5: Especificações Módulo Relé 1 Canal 5v

TENSÃO DE OPERAÇÃO	5V DC (VCC e GND)
TENSÃO DE SINAL	TTL - 5V DC (IN)
CORRENTE TÍPICA DE OPERAÇÃO	80 mA
O RELÉ POSSUI	contato NA e NF
CAPACIDADE DO RELÉ	30 V DC e 10A ou 250V AC e 10A
TEMPO DE RESPOSTA	5~10ms
POSSUI	Indicador LED de funcionamento
DIMENSÕES	43mm (L) x 17mm (C) x 19mm (H)

Fonte: Elaborado pelo autor, 2024

2.6 Bomba de água

As bombas de água são componentes essenciais em diversos setores, desempenhando um papel crucial tanto na indústria quanto na agricultura. Elas são responsáveis por garantir o abastecimento hídrico em diferentes processos e aplicações, tornando-se indispensáveis para a manutenção de operações eficientes e

produtivas. No contexto agrícola, sua utilização é fundamental para a irrigação, assegurando que as culturas recebam a quantidade adequada de água para um desenvolvimento saudável (Livia, 2024).

O funcionamento das bombas de água baseia-se em princípios de mecânica e hidráulica, convertendo energia mecânica em energia hidráulica para movimentar a água. Esse processo ocorre por meio de um motor que aciona um rotor, responsável por gerar o fluxo de água através de um sistema de tubulações e válvulas (Livia, 2024).

Figura 7: Minibomba de Água para Arduino



Fonte: Usinainfo, 2021

2.5.1 Mini Bomba Submersível

As mini bombas submersíveis, assim como as bombas tradicionais, são dispositivos que convertem energia para a movimentação de líquidos. Nesse caso, transformam energia elétrica em energia cinética, utilizando um mecanismo rotacional para impulsionar a água. Esse mecanismo consiste em uma pá giratória acionada por um motor, permitindo a transferência de líquidos de um local para outro. Dependendo do modelo, a fonte de energia pode ser elétrica, pneumática ou até mesmo manual (ALVES, 2023).

Para garantir a integridade do equipamento e seu correto funcionamento, a bomba nunca deve ser acionada a seco, pois a ausência de líquido pode comprometer seus componentes internos e reduzir sua vida útil.

Equipada com um motor compacto e eficiente, a mini bomba de água é capaz de movimentar entre 80 e 120 litros por hora, garantindo um fluxo contínuo e confiável. Seu design permite operação submersa sem comprometer a segurança ou eficiência do sistema. Para um desempenho adequado, é essencial que a bomba esteja completamente submersa durante a operação. Neste modelo específico, a saída de água está posicionada a 90 graus em relação à entrada, e o eixo do motor permanece em contato direto com a água durante seu funcionamento.

No contexto deste trabalho, a mini bomba submersível é responsável pelo transporte da água do reservatório até o solo da planta. Acionada por um módulo relé de 5V, a bomba entra em funcionamento sempre que o sensor de umidade detecta níveis abaixo do ideal, garantindo que a irrigação ocorra de forma automática.

Figura 8: Mini Bomba Submersível p/ Água



Fonte: Eletrogate, 2024

Tabela 6: Especificações Mini Bomba Submersível p/ Água

MODELO	JT100
TENSÃO ADEQUADA	DC 3 a 5 V
CORRENTE MÁXIMA	~1.5 A
ELEVAÇÃO MÁXIMA	1 m
VAZÃO DE ÁGUA	~1 a 1,5 l/m
COMPRIMENTO DO FIO	7 cm
DIÂMETRO DA ENTRADA	~3,3 mm
DIÂMETRO DA SAÍDA	~4,5 mm
DIMENSÕES (CXLXA)	43,5x23x30 mm
PESO	26 g

Fonte: Elaborado pelo autor, 2024

2.7 Buzzer

Buzzer é um componente eletrônico que é composto por 2 camadas de Metal e uma terceira camada interna de cristal Piezoelétrico, este componente recebe uma fonte de energia e através dela emite uma frequência sonora (SALES; DA SILVA; AMADEU, 2017).

Existem dois tipos principais de buzzers:

Buzzer Ativo: Os buzzers ativos são dispositivos que não necessitam de circuitos externos para gerar som. Basta fornecer a alimentação apropriada para que o circuito interno do buzzer ative as vibrações no diafragma, produzindo som. Isso torna seu uso extremamente simples e sem complicações. No entanto, uma desvantagem dos buzzers ativos é que eles não podem produzir sons diferentes, pois sua frequência de operação é fixa.

Buzzer Passivo: Diferente do buzzer ativo, o buzzer passivo necessita de um circuito externo para gerar som. A frequência do som dependerá das características desse circuito externo. Isso permite que o buzzer passivo produza múltiplos sons, como em uma melodia, tornando-o uma ótima escolha para projetos que requerem diversidade sonora.

2.7.1 Buzzer Ativo 5v

No projeto de irrigação automatizado para cultivo residencial de tomates, foi utilizado um Buzzer Ativo 5V. Este dispositivo é projetado para emitir sons agudos ao receber uma tensão de 5V, sem a necessidade de um circuito de acionamento externo. Quando energizado, o buzzer gera um sinal sonoro contínuo e de frequência fixa, facilitando a criação de alertas sonoros. No contexto deste trabalho, o Buzzer Ativo 5V é utilizado para emitir um alerta sonoro quando o nível de água no reservatório está crítico, garantindo que o usuário seja informado imediatamente sobre a necessidade de reabastecimento. A simplicidade de uso do buzzer ativo, que requer apenas uma conexão direta ao microcontrolador Arduino, contribui para a eficiência e praticidade do sistema.

Tabela 7: Especificação Buzzer ativo 5V

TENSÃO DE OPERAÇÃO	3,5 - 5V
TIPO	Beep contínuo
COR	preto
DIÂMETRO	12mm
ALTURA	10mm

Fonte: Elaborado pelo autor, 2024

2.8 Arduino IDE

O Para implementar programas para serem utilizados em um Arduino é necessário utilizar uma IDE, onde todo o código é escrito, são feitos testes para encontrar possíveis erros e para transferir o programa para a placa. Esse ambiente de programação pode ser baixado de forma gratuita no site oficial da plataforma (DA SILVA ROCHA, 2019).

A linguagem de programação utilizada no Arduino IDE é baseada na linguagem C++, sendo uma linguagem simples e compreensível. Alguns conceitos básicos incluem declaração de variáveis, estruturas condicionais (como if, else), estruturas de repetição (como for e while) e operações matemáticas(DA SILVA ROCHA, 2019).

Este software proporciona acesso a um leque de funcionalidades essenciais para a criação, análise e implementação de códigos em sua placa Arduino.

Dentro deste espectro, a IDE facilita o processo através de recursos como:

- Editor de código para escrita;
- Compilador para transformar o código;
- Uma vasta biblioteca de funções pré-estabelecidas;
- Gerenciamento de diferentes modelos de placas;
- Monitoramento serial para comunicação;
- Ferramentas dedicadas à depuração;
- Funcionalidade para o carregamento eficiente de códigos.

Adicionalmente, no decorrer do desenvolvimento de um projeto, a Arduino IDE é capaz de notificar sobre quaisquer inconformidades na configuração ou no código,

apontando também a possível localização do erro, facilitando assim a correção e otimização do projeto.

2.8.1 Download do Arduino IDE.

Este ambiente de desenvolvimento destaca-se pela sua compatibilidade abrangente com praticamente todos os sistemas operacionais, permitindo a instalação gratuita a partir do site oficial [arduino.cc](https://www.arduino.cc).

O programa está disponível para ser baixado em versões para Windows, Mac OS e Linux, embora existam certos pré-requisitos específicos:

- Para Windows, é necessário ter a versão 10 ou superior, com um processador de 64 bits.
- Usuários de Linux também precisam de um processador com arquitetura de 64 bits.
- No Mac OS, é exigida a versão 10.14 ("Mojave") ou mais recente, igualmente com suporte a 64 bits.

Portanto, para a instalação e uso efetivo, seu computador deve estar alinhado a esses requisitos operacionais. Além disso, é indispensável possuir uma placa Arduino e um cabo USB compatível para a realização de projetos, uma vez que a placa Arduino é essencial para a compilação e transferência do código desenvolvido no software por meio do cabo USB.

Figura 9: Logo Arduino



Fonte: [Arduino.cc](https://www.arduino.cc), 2024

3 METODOLOGIA

Para o desenvolvimento do sistema, foram realizadas pesquisas na área e uma análise de sistemas similares já existentes. Com base nas informações coletadas, o projeto do sistema de irrigação automatizado foi desenvolvido. Esse processo envolveu a definição da arquitetura do sistema, a escolha dos componentes, bem como a integração e implementação deles para o funcionamento conforme os objetivos propostos.

3.1 Visão geral do projeto

O projeto tem como objetivo desenvolver um sistema de irrigação automatizado residencial para o cultivo de tomates, utilizando sensores de umidade do solo, sensor de nível de água e um microcontrolador Arduino. O sistema foi projetado para monitorar continuamente a umidade do solo e a disponibilidade de água no reservatório, acionando a irrigação sempre que necessário.

A umidade do solo é monitorada por um sensor de umidade instalado próximo às raízes, que envia os dados para o Arduino. Além disso, um sensor de nível de água é utilizado para verificar se há água suficiente no reservatório. Quando o sensor de umidade detecta que a umidade do solo está abaixo da faixa ideal para o cultivo de tomates (50% - 70%), o microcontrolador verifica o nível de água disponível. Caso o reservatório possua água suficiente, um módulo relé é acionado, ligando uma mini bomba submersível para iniciar a irrigação.

A irrigação ocorre por meio de uma mangueira posicionada sobre o solo, liberando a água diretamente na base da planta, caracterizando um método de irrigação por superfície. O ciclo de irrigação continua até que o sensor de umidade registre que o solo atingiu um nível adequado. Em seguida, a bomba é desligada automaticamente pelo relé, retornando o sistema ao estado de monitoramento.

Para indicar visualmente o nível de umidade do solo, o sistema utiliza LEDs indicativos, com as seguintes cores:

- LED vermelho: Indica umidade baixa (menos de 50%).
- LED amarelo: Indica umidade moderada (entre 50% e 60%).
- LED verde: Indica umidade dentro da faixa ideal (acima de 60%).

Além disso, um buzzer é ativado para alertar o usuário quando o nível do reservatório está baixo e a irrigação não pode ser realizada. Dessa forma, o sistema desenvolvido com Arduino proporciona um método simples e eficiente de automação da irrigação, garantindo que as plantas recebam água de maneira consistente e adequada, sem a necessidade de monitoramento manual constante.

3.1.1 Lista dos principais componentes utilizados.

- 1 Arduino Uno Rev3
- 1 Mini Bomba Submersível
- 1 Módulo Sensor De Umidade De Solo
- 1 Módulo de relé 5V
- 1 fonte 5V utilizada na bomba d'água.
- 1 fonte de 12V para alimentar o Arduino.
- 1 mangueira utilizada na bomba d'água.
- Jumpers
- 1 mini Protoboard
- LEDs (3 unidades: vermelho, amarelo, verde)
- Sensor de Nível de Água 90°
- Buzzer Ativo 5v
- Resistores

3.2 Estrutura do projeto

A montagem do hardware do sistema de irrigação automatizado envolve a integração cuidadosa dos seguintes componentes, conforme descrito no esquema de design:

- **Arduino Uno Rev3:** Este microcontrolador atua como o cérebro do sistema, processando as informações recebidas dos sensores de umidade do solo e nível de água, e acionando o módulo de relé conforme necessário.

- **Mini Bomba Submersível:** Utilizada para mover a água do reservatório para os vasos de plantas. A escolha de uma bomba submersível garante um funcionamento eficiente e silencioso, ideal para ambientes residenciais.
- **Módulo Sensor de Umidade de Solo:** Inserido no solo dos vasos de plantas, este sensor monitora constantemente o nível de umidade, fornecendo dados precisos para o Arduino decidir quando ativar a bomba de água.
- **Módulo de Relé 5V:** Serve como uma ponte entre o Arduino e a mini bomba de água. Quando o Arduino determina que a irrigação é necessária, ele envia um sinal para o módulo relé, que por sua vez ativa a bomba de água.
- **Sensor de Nível de Água:** Este sensor detecta o nível de água no reservatório, garantindo que a bomba de água não funcione a seco. Ele envia um sinal ao Arduino quando o nível de água está baixo, desencadeando um alerta.
- **Fonte de 5V:** Utilizada para alimentar a mini bomba submersível de água. É crucial que a fonte forneça a tensão adequada para o funcionamento eficaz da bomba.
- **Fonte de Alimentação para o Arduino:** Necessária para alimentar o Arduino Uno, garantindo que o microcontrolador tenha energia suficiente para executar suas funções sem interrupções.
- **Mangueira:** Conecta a saída da mini bomba submersível aos vasos de plantas, direcionando o fluxo de água conforme necessário. A escolha de uma mangueira adequada é importante para garantir uma distribuição eficiente da água.
- **LEDs (Vermelho, Amarelo, Verde):** Utilizados para indicar os níveis de umidade do solo. O LED vermelho indica umidade baixa, o amarelo indica umidade média e o verde indica umidade alta.
- **Buzzer:** Emite um alarme sonoro quando o nível de água no reservatório está crítico, alertando o usuário sobre a necessidade de reabastecimento.

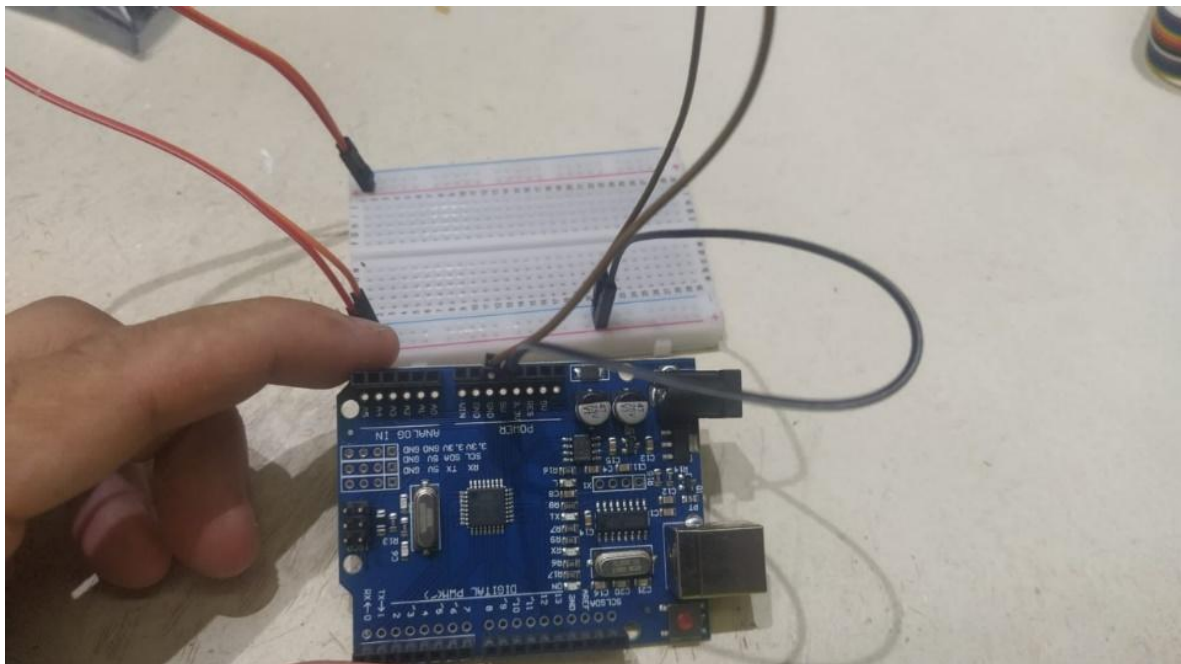
3.3 Montagem do protótipo

A construção do protótipo será realizada em várias etapas, descritas a seguir.

3.3.1 Ligações eletrônicas

Para a montagem do protótipo, optou-se pela utilização de um protoboard, visando maior organização e facilidade na execução das conexões. Inicialmente, os 5 volts e o GND (terra) do Arduino foram conectados a duas trilhas distintas no protoboard, utilizando-se jumpers macho-macho. Esta etapa estabelece as linhas de alimentação e terra que serão compartilhadas pelos componentes do circuito.

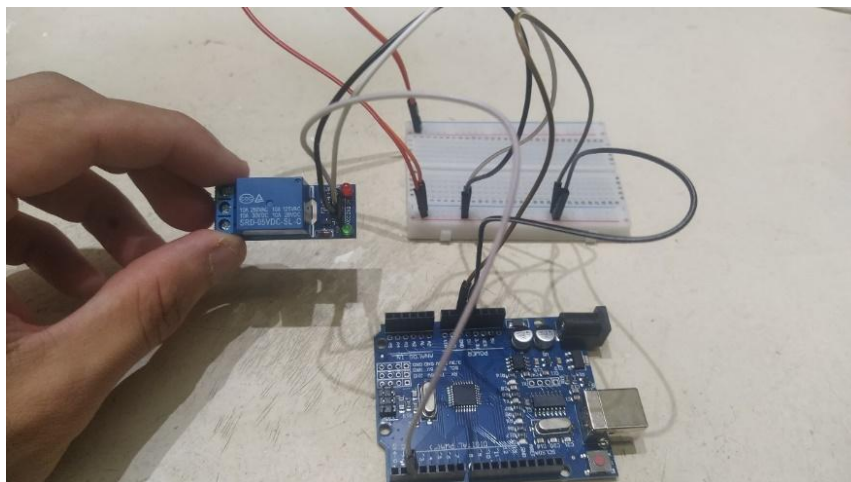
Figura 10: Ligação 5v e GND Arduino



Fonte: Elaborado pelo autor, 2024

Posteriormente, as conexões do módulo de relé foram realizadas. Utilizando jumpers macho-fêmea, o VCC e o GND do módulo de relé foram ligados às trilhas de 5 volts e GND no protoboard, respectivamente, garantindo a alimentação do relé diretamente pelos 5 volts fornecidos pelo Arduino. O pino de controle (IN) do relé foi conectado ao pino digital 7 do Arduino, configurado como saída digital para o acionamento do relé. O relé atua como um interruptor controlado eletricamente, permitindo a ativação e desativação de dispositivos conectados a ele.

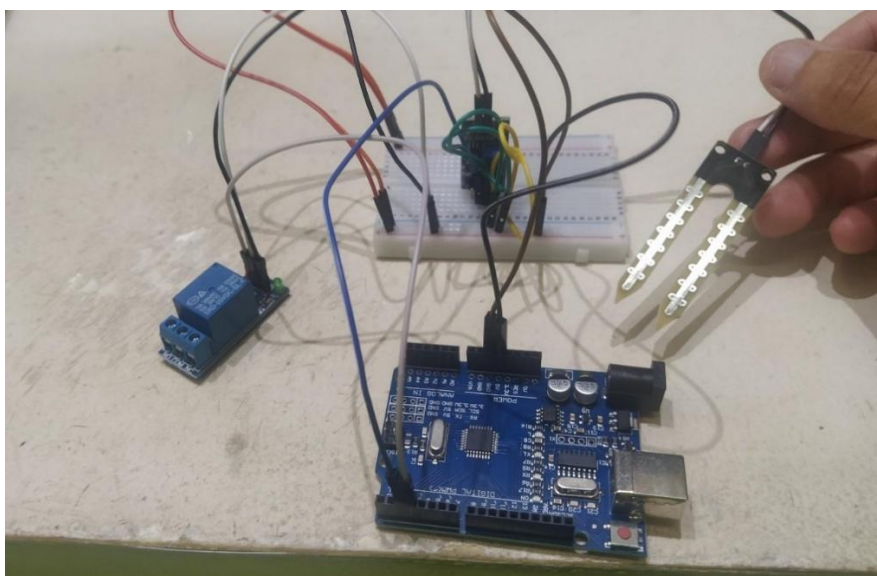
Figura 11: Ligação relé no Arduino



Fonte: Elaborado pelo autor, 2024

Para o módulo sensor de umidade do solo, sua conexão também foi efetuada via protoboard, mantendo a organização do circuito. Os jumpers foram conectados às respectivas saídas do sensor (VCC, GND) e dirigidos às trilhas de alimentação e terra no protoboard. A saída analógica do sensor foi conectada ao pino analógico A0 do Arduino, permitindo a leitura dos sinais de umidade do solo.

Figura 12: Ligação Sensor de Umidade



Fonte: Elaborado pelo autor, 2024

O sensor de nível de água foi conectado utilizando jumpers, ligando um terminal ao pino digital 8 do Arduino e o outro terminal ao GND no protoboard. Este sensor

permite a detecção do nível de água no reservatório, enviando um sinal ao Arduino quando o nível de água está baixo.

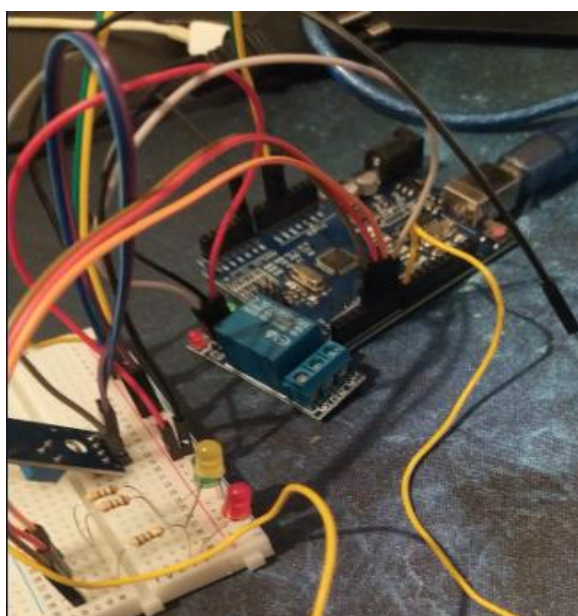
Figura 13: Ligação Sensor de Nível de Água



Fonte: Elaborado pelo autor, 2024

Os LEDs (vermelho, amarelo e verde) foram conectados aos pinos digitais 4, 5 e 6 do Arduino, respectivamente, através de resistores para limitar a corrente. Os cátodos dos LEDs foram conectados ao GND no protoboard, enquanto os ânodos foram conectados aos pinos digitais correspondentes.

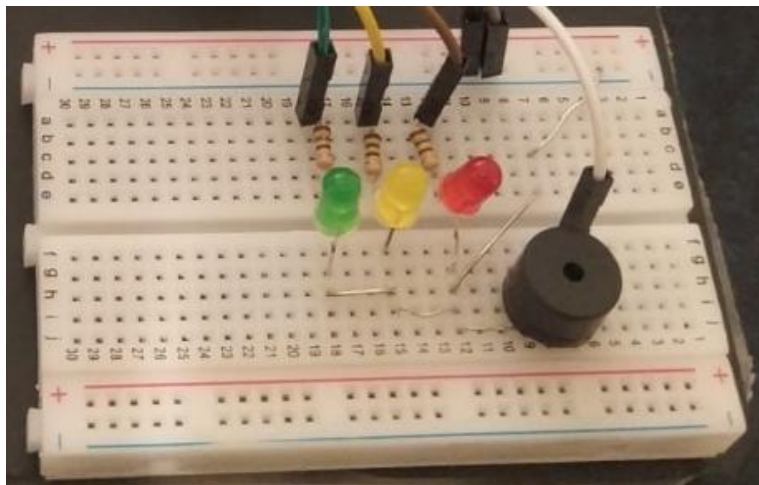
Figura 14: Ligação dos LEDs



Fonte: Elaborado pelo autor, 2024

O buzzer foi conectado ao pino digital 9 do Arduino, com um resistor em série para limitar a corrente. O terminal negativo do buzzer foi conectado ao GND no protoboard.

Figura 15: Ligação do Buzzer



Fonte: Elaborado pelo autor, 2024

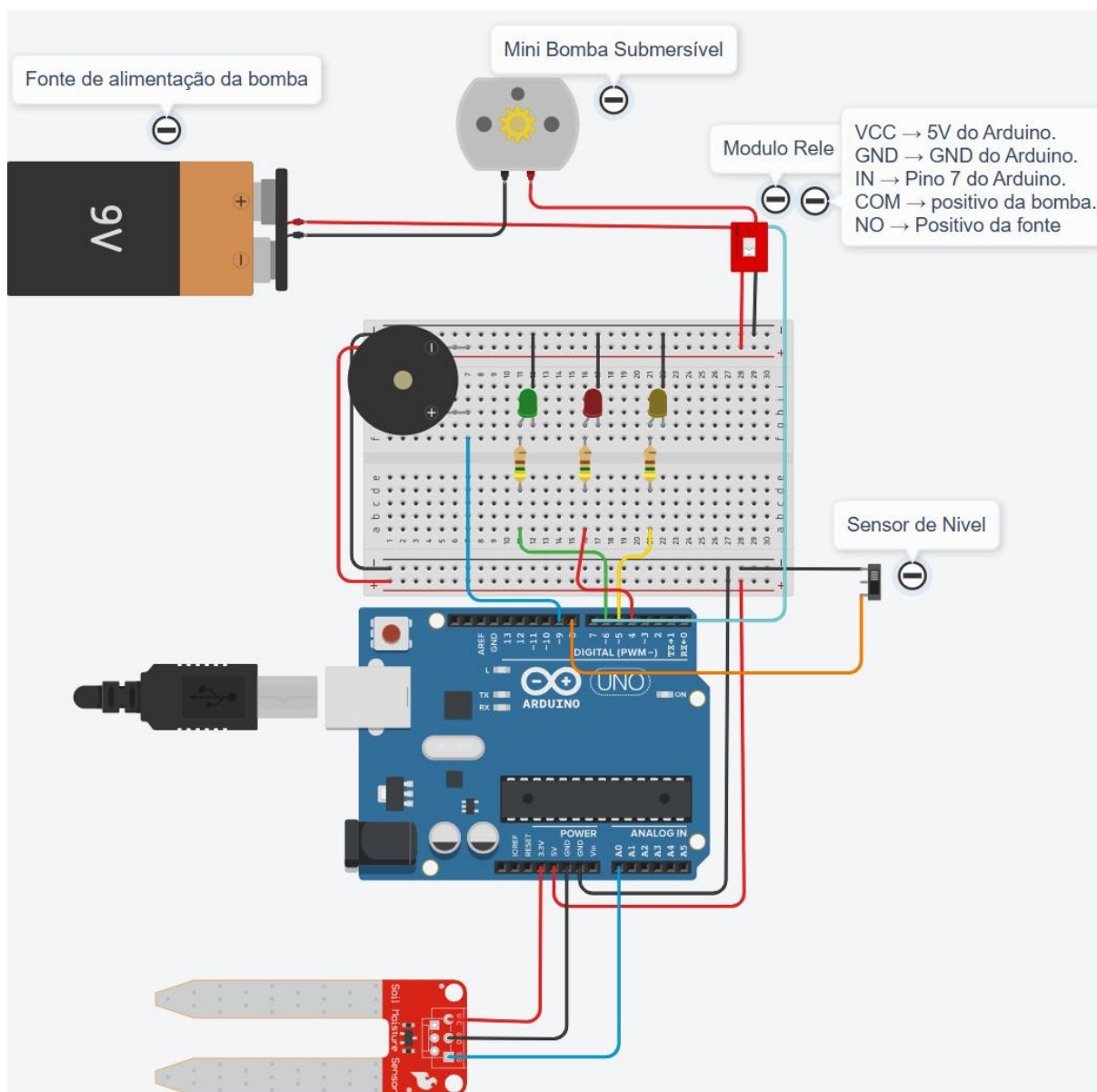
Finalmente, a conexão da mini bomba submersível foi realizada através do módulo de relé. A bomba foi alimentada por uma fonte de 5 volts externa, adequada para seu funcionamento. O fio positivo da bomba d'água foi conectado ao terminal Normalmente Aberto (NO) do relé, enquanto o fio negativo da bomba foi diretamente ligado ao negativo (GND) da fonte de 5 volts. Assim, quando o relé é acionado pelo Arduino (via detecção de baixa umidade pelo sensor), o circuito é fechado e a bomba é ativada, iniciando a irrigação. Quando o relé é desativado, o circuito se abre e a bomba é desligada, cessando a irrigação.

Esta configuração assegura um controle eficiente do sistema de irrigação, permitindo que ele opere automaticamente de acordo com a umidade do solo, sem a necessidade de intervenção manual constante.

3.3.2 Diagrama do Circuito Elétrico

Para melhor visualização do sistema de irrigação automatizado, foi elaborado um diagrama elétrico utilizando a plataforma Tinkercad. O circuito foi desenvolvido considerando todos os componentes essenciais do projeto, incluindo o microcontrolador, sensores, atuadores e indicadores visuais e sonoros.

Figura 16: Diagrama do Circuito Elétrico no Tinkercad.

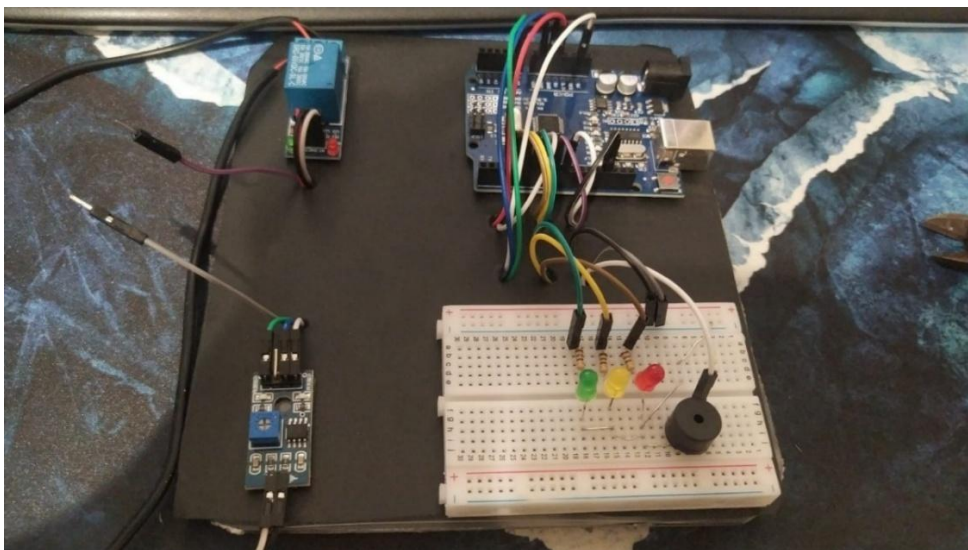


Fonte: Elaborado pelo autor, 2024

3.3.3 Estrutura mecânica

Para a estrutura mecânica do projeto, foi utilizado um acrílico transparente como base, proporcionando uma aparência moderna e organizada ao protótipo. Todos os componentes foram fixados na superfície do acrílico utilizando fita dupla face, garantindo uma fixação firme e segura. Além disso, para melhorar a estética do protótipo, foi aplicado um papel crepom preto sobre a placa de acrílico.

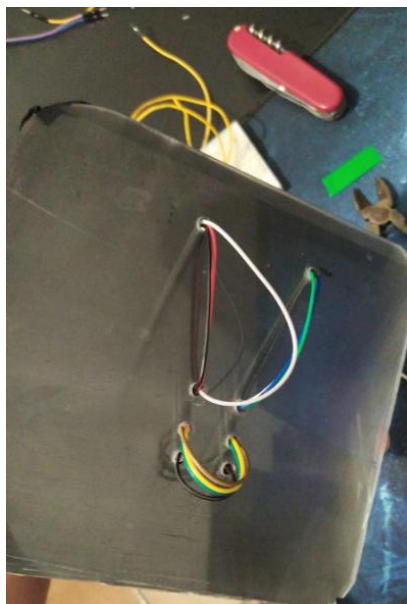
Figura 17: Base de Acrílico com Componentes.



Fonte: Elaborado pelo autor, 2024

Para facilitar a organização dos cabos, foram feitos furos estratégicos no acrílico, permitindo que os fios passassem por baixo da base de maneira ordenada.

Figura 18: Organização dos Cabos



Fonte: Elaborado pelo autor, 2024

O reservatório de água foi confeccionado a partir de um pote branco, que acomoda tanto a mini bomba submersível quanto o sensor de nível de água. As conexões da bomba e do sensor foram cuidadosamente organizadas dentro do pote, garantindo que todo o sistema funcionasse de maneira eficiente e sem obstruções..

Figura 19: Reservatório de Água



Fonte: Elaborado pelo autor, 2024

Essa montagem melhorou a organização do trabalho e facilitou a manutenção e o ajuste dos componentes.

3.3.4 Programação do sistema - Arduino IDE

Todo processo de programação se encontra no apêndice A e consiste na implementação de um script simples, porém eficaz, destinado ao microcontrolador Arduino, para automatizar o processo de irrigação com base na umidade do solo e no nível de água do reservatório. O código é desenvolvido e carregado utilizando a plataforma Arduino IDE.

Inicialmente, são definidas várias constantes para representar os pinos de conexão dos componentes ao Arduino:

- AnalogPin, associado ao pino analógico A0, conecta-se ao módulo sensor de umidade do solo.
- low, medium e high, associados aos pinos digitais 4, 5 e 6, respectivamente, controlam os LEDs indicativos de níveis de umidade do solo.
- Rele, vinculado ao pino digital 7, controla a mini bomba submersível por meio de um módulo de relé.
- levelSensor, associado ao pino digital 8, conecta-se ao sensor de nível de água.
- buzzer, vinculado ao pino digital 9, é utilizado para emitir alertas sonoros.

Essas constantes são utilizadas para facilitar a leitura do código, permitindo rápidas alterações dos pinos de conexão, caso necessário.

São declaradas variáveis para armazenar as leituras dos sensores e controlar o estado do sistema:

- AnalogIn, que armazena a leitura do sensor de umidade do solo.
- Variáveis booleanas on e level, inicialmente configuradas como false, que armazenam o estado de funcionamento do sistema de irrigação e o estado do nível de água, respectivamente.
- Na função setup(), configura-se os modos dos pinos conforme seus propósitos:
- O pino conectado ao relé é configurado como OUTPUT, para enviar sinais ao relé.
- Os pinos conectados aos LEDs e ao buzzer são definidos como OUTPUT.
- O pino conectado ao sensor de nível de água é definido como INPUT_PULLUP, permitindo a leitura dos sinais de nível de água.

Além disso, estabelece-se o estado inicial do relé como desligado HIGH, considerando a característica do módulo de relé utilizado, que é ativado em nível lógico baixo LOW.

Dentro da função loop(), realiza-se continuamente as seguintes operações:

Verificação do Nível de Água:

- Se o nível de água estiver baixo (digitalRead(levelSensor) == HIGH) e o sistema de irrigação estiver ligado (on == true), o sistema entra em modo de emergência.
- O relé é desligado HIGH, a bomba de água é parada e o buzzer é ativado para emitir um alarme sonoro por 3 segundos. Em seguida, o sistema de irrigação é desligado definindo on como false.

Leitura do Sensor de Umidade:

- O valor do sensor de umidade do solo é lido e mapeado para uma escala de 0 a 100%.
- O valor de umidade é exibido no monitor serial para depuração.

Controle dos LEDs Indicativos:

- Umidade baixa: Acende o LED vermelho (low).
- Umidade média: Acende o LED amarelo (medium).
- Umidade alta: Acende o LED verde (high).

Controle do Relé:

- Se a umidade estiver abaixo do nível ideal ($\leq 50\%$) e o sistema de irrigação estiver desligado (on == false), o nível de água é verificado.
- Se o nível de água estiver adequado (digitalRead(levelSensor) == LOW), o relé é ligado LOW, a bomba de água é ativada e o sistema de irrigação é ligado definindo on como true.

Este processo se repete indefinidamente, garantindo que as plantas sejam irrigadas automaticamente sempre que a umidade do solo estiver abaixo do nível ajustado pelo sensor de umidade e que o nível de água seja adequado.

Após a finalização da codificação, o script é carregado para o Arduino conectado via USB ao computador. No caso em questão, o Arduino Uno está conectado na porta COM3. A seleção do modelo de placa correto e a porta de comunicação apropriada são essenciais para o sucesso do upload do código para o microcontrolador.

4 MELHORIAS

Com a implementação do projeto inicial, foram identificadas diversas oportunidades de aprimoramento, tanto na estrutura física quanto na lógica do sistema de irrigação. Essas melhorias visam otimizar o funcionamento do sistema, tornando-o mais versátil e proporcionando um maior controle ao usuário. Além disso, aprimoramentos na interface e na conectividade podem agregar novas funcionalidades, ampliando a aplicabilidade do sistema.

4.1 ESP32

O ESP32, lançado em 2016 como sucessor do ESP8266, representa uma evolução significativa para projetos de automação e Internet das Coisas (IoT) devido ao seu custo acessível e à vasta disponibilidade de recursos de programação. Este microcontrolador é amplamente adotado em aplicações móveis, dispositivos eletrônicos portáteis e sistemas IoT, graças à sua capacidade de comunicação sem fio dual (Wi-Fi e Bluetooth) e ao seu consumo energético otimizado (SOARES, 2023).

Características Técnicas do ESP32:

- CPU: Xtensa® dual-core 32-bit LX6;
- Memória RAM: 520 KB;
- Memória ROM: 448 KB;
- Memória Flash: 4MB;
- Wi-Fi 2,4 GHz;
- Bluetooth BLE 4.2.

Figura 20: ESP32



Fonte: Eletrogate, 2024

A substituição do Arduino Uno pelo ESP32 no sistema de irrigação automatizado proporcionaria uma maior flexibilidade, permitindo a implementação de funcionalidades avançadas e a integração com redes IoT para monitoramento e controle remoto.

4.2 Sensor ultrassônico

A adição de um sensor ultrassônico ao sistema oferece uma camada adicional de monitoramento, especificamente para medir o nível de água no reservatório de forma mais precisa e eficiente.

Os sensores ultrassônicos operam emitindo ondas sonoras em frequências acima da faixa audível humana (aproximadamente 40 kHz). Essas ondas são refletidas por objetos ou obstáculos no caminho e retornam ao sensor, permitindo a medição da distância com base no tempo de retorno do sinal (REIS, 2024).

. O transdutor do sensor atua como um microfone para receber e enviar o som ultrassônico. O sensor determina a distância até um alvo medindo os lapsos de tempo entre o envio e o recebimento do pulso ultrassônico (REIS, 2024).

Figura 21: Sensor ultrassônico



Fonte: Eletrogate, 2024

A inclusão do sensor ultrassônico no sistema de irrigação automatizado aprimora significativamente o monitoramento do reservatório de água, contribuindo para uma gestão mais eficiente dos recursos hídricos.

4.3 Telegram

Integrar o sistema de irrigação com a plataforma Telegram possibilitaria o monitoramento e controle remoto através de mensagens e comandos enviados pelo aplicativo. Esta funcionalidade pode ser expandida com o uso de chatbots, que automatizam respostas e notificações, oferecendo uma interface amigável e acessível para os usuários.

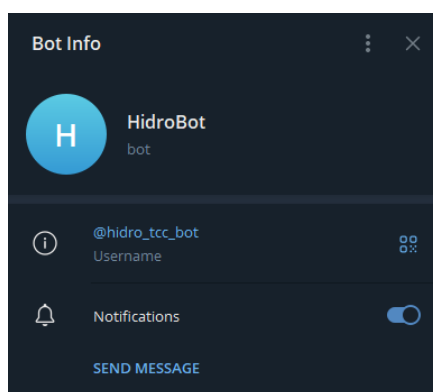
4.3.1 Chatbot

Os chatbots são programas de computador projetados para interagir com usuários por meio de linguagem natural, permitindo a automação de tarefas e proporcionando uma experiência mais fluida e intuitiva na comunicação entre humanos e máquinas (DE ARAÚJO, 2022).

No contexto deste projeto, um chatbot foi implementado para permitir que o usuário acompanhe, em tempo real, o funcionamento do sistema de irrigação automatizado e receba notificações sobre o estado da umidade do solo e do nível do reservatório.

O chatbot utilizado neste projeto foi integrado ao aplicativo **Telegram**, permitindo que o usuário receba informações sobre a umidade do solo e o estado da irrigação diretamente no celular. O bot responde a comandos pré-definidos e envia notificações automáticas sempre que ocorre uma mudança no sistema. Dessa forma, é possível acompanhar e controlar remotamente a irrigação das plantas.

Figura 22: HidroBot Telegram



Fonte: Elaborado pelo autor, 2024

4.4 Programação do novo sistema.

O processo de programação do sistema de irrigação automatizado foi atualizado para incluir funcionalidades avançadas, como monitoramento remoto via Telegram e controle do sistema utilizando o microcontrolador ESP32. Todo o código, disponível no apêndice B, foi desenvolvido e carregado utilizando a plataforma Arduino IDE.

Funcionamento geral:

- O sistema monitora a umidade do solo e o nível de água no reservatório.
- Se a umidade do solo estiver baixa e houver água suficiente no reservatório, o sistema ativa a bomba para irrigar o solo.
- O sistema se comunica com o usuário via Telegram, enviando alertas sobre o estado do solo e do reservatório.
- LEDs indicam visualmente o estado do solo, enquanto o buzzer alerta para situações críticas.

Este código permite um controle eficiente e interativo do sistema de irrigação, aumentando a conveniência e a eficácia do monitoramento e controle remoto do sistema.

4.5 Estrutura mecânica

Com as melhorias implementadas no sistema, foi desenvolvida uma nova estrutura mecânica para acomodar todos os componentes de maneira organizada e funcional, garantindo a execução do método de irrigação por superfície adotado no projeto. A estrutura é composta por placas de acrílico cortadas e montadas em formato de caixa, proporcionando um suporte adequado para os principais elementos do sistema: o reservatório de água, a parte lógica (circuitos e microcontrolador) e a planta.

Para a confecção da estrutura, foram utilizados acrílicos de cores diferentes, otimizando tanto a estética quanto a funcionalidade do sistema. A base e as laterais foram produzidos em acrílico preto, conferindo robustez e proteção aos componentes internos. Já a tampa foi confeccionada em acrílico fosco, permitindo uma visualização parcial do interior sem comprometer a segurança dos circuitos eletrônicos.

A mini bomba submersível transporta a água do reservatório até a planta por meio de uma mangueira direcionada para a superfície do vaso, garantindo que a irrigação ocorra de forma controlada e uniforme.

Outro benefício significativo dessa estrutura é a sua portabilidade, possibilitando que o usuário transporte o sistema com facilidade e o posicione em diferentes ambientes conforme sua necessidade.

Figura 23: Estrutura - Visão lateral



Fonte: Elaborado pelo autor, 2024

Figura 24: Estrutura - Visão de cima



Fonte: Elaborado pelo autor, 2024

Figura 25: Estrutura - Saída dos cabos de alimentação



Fonte: Elaborado pelo autor, 2024

5 RESULTADOS

Este capítulo destina-se a apresentar os resultados obtidos após a montagem e o teste do sistema de irrigação automática. A análise foca principalmente no funcionamento prático do sistema e na avaliação do desempenho em relação aos objetivos iniciais estabelecidos.

5.1 Funcionamento do Sistema com Arduino

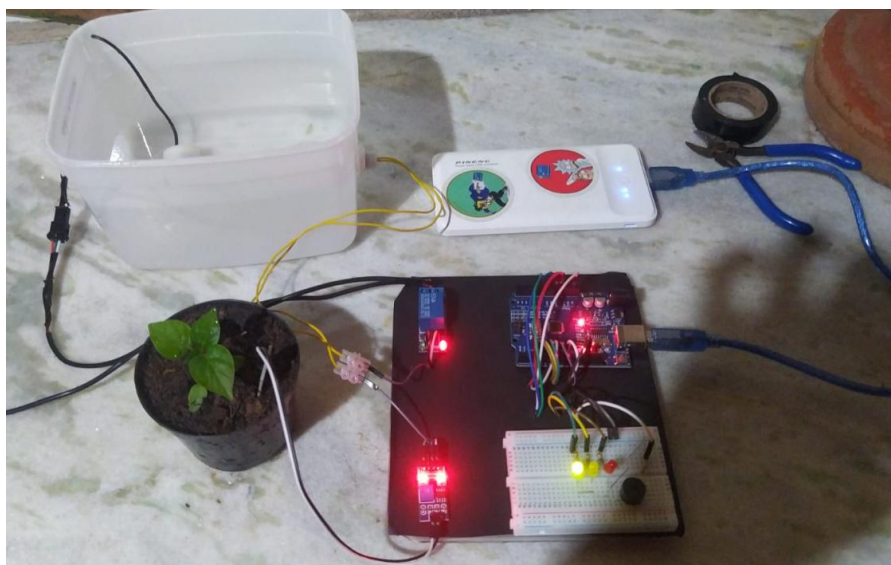
Após a conclusão da montagem, o sistema foi submetido a uma série de testes para avaliar sua funcionalidade. Os resultados demonstraram que o sensor de umidade do solo foi capaz de monitorar adequadamente o nível de umidade e transmitir esses dados ao Arduino Uno Rev3. Com base nesses dados, o Arduino atuou conforme o esperado, acionando o módulo relé e, por sua vez, ativando a mini bomba submersível para irrigar os vasos.

As faixas de umidade definidas no sistema foram fundamentais para o seu funcionamento:

- Umidade baixa (menos de 50%): Indicada pelo LED vermelho.
- Umidade média (entre 50% e 60%): Indicada pelo LED amarelo.
- Umidade alta (mais de 60%): Indicada pelo LED verde.

Quando a umidade do solo caiu abaixo de 50%, o sistema acionou a bomba de água para iniciar a irrigação. A bomba funcionou dentro da faixa de capacidade projetada, o que se mostrou adequado para o volume dos vasos utilizados nos testes. Após a umidade atingir níveis ideais, a bomba foi desligada automaticamente, demonstrando a eficiência e precisão do sistema em manter os níveis de umidade dentro das faixas desejadas.

Figura 26: Primeiro teste do sistema



Fonte: Elaborado pelo autor, 2024

5.1.1 Desempenho do Sensor de Umidade

Os valores de umidade foram lidos pelo sensor e convertidos em porcentagens, o que facilitou a visualização clara e imediata do estado do solo. Esta conversão permitiu ao sistema tomar decisões informadas sobre quando ativar ou desativar a irrigação. Quando a umidade do solo caiu abaixo do pré-estabelecido de 50%, o sistema prontamente acionou a mini bomba submersível, iniciando a irrigação.

A capacidade do sensor de detectar rapidamente a recuperação dos níveis de umidade foi satisfatória. Assim que a umidade do solo atingiu os níveis ideais, a bomba de água foi automaticamente desligada. Este comportamento cíclico do sensor e do sistema de controle demonstrou sua capacidade de manter os níveis de umidade do solo dentro da faixa desejada, evitando tanto a sub-irrigação quanto a sobre-irrigação.

O desempenho consistente do sensor de umidade foi fundamental para o sucesso do sistema de irrigação automatizado, destacando-o como o componente mais crítico do projeto.

5.1.2 Funcionamento do Sensor de Nível de Água

Durante os testes, o sensor detectou quando o nível de água no reservatório estava baixo, acionando um alarme sonoro através do buzzer. Este alerta permitiu intervenções rápidas para reabastecimento do reservatório, garantindo que o sistema continuasse operando sem interrupções.

5.1.3 Atuação do Relé e Bomba de Água

O módulo relé funcionou de maneira confiável, respondendo aos comandos do Arduino sem atrasos perceptíveis. A mini bomba submersível, controlada pelo relé, mostrou-se eficiente em entregar a quantidade necessária de água para os vasos. Os testes confirmaram que o relé e a bomba poderiam operar repetidamente sem falhas, mantendo o solo adequadamente irrigado.

5.1.4 Indicadores Visuais e Sonoros

Para auxiliar o monitoramento presencial do sistema, foram implementados LEDs indicadores e um buzzer para alertas sonoros. Os LEDs foram utilizados para fornecer um feedback visual imediato sobre o nível de umidade do solo:

- LED vermelho: Indica umidade baixa, sinalizando a necessidade de irrigação.
- LED amarelo: Indica umidade intermediária, dentro da faixa média aceitável.
- LED verde: Indica umidade alta, mostrando que o solo está em boas condições.

Além disso, o buzzer foi ativado em situações específicas, como quando o nível de água no reservatório atingia um estado crítico. Isso permitiu que o usuário identificasse rapidamente a necessidade de reabastecimento do reservatório e evitasse falhas no funcionamento da bomba.

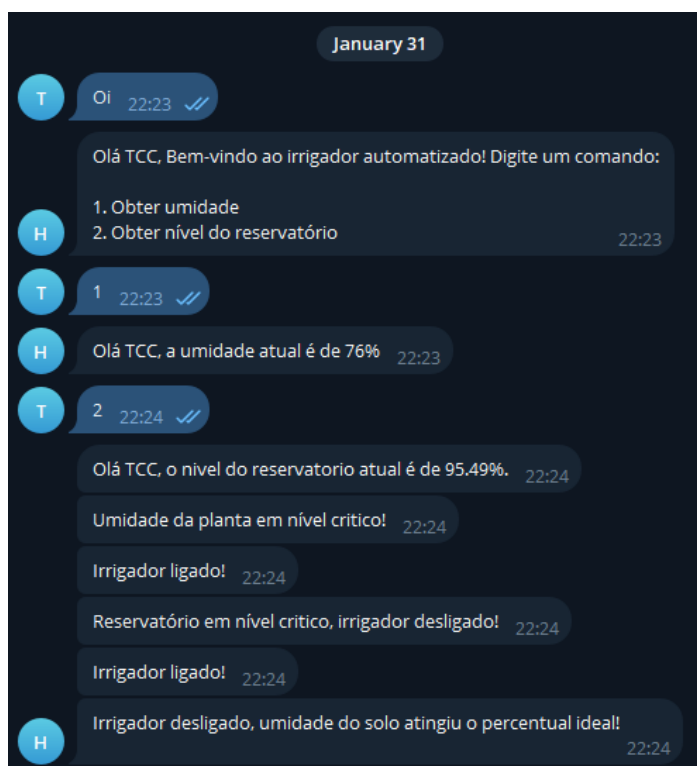
5.2 Funcionamento do Sistema após melhorias

Após a implementação das melhorias, o sistema de irrigação foi aprimorado para oferecer maior eficiência, funcionalidades adicionais e um controle mais preciso dos processos.

5.2.1 ESP32 e Monitoramento Remoto

Durante os testes, o sistema foi capaz de enviar atualizações sobre a umidade do solo e o nível de água diretamente para o chat do Telegram. O usuário pôde, por exemplo, enviar comandos para obter a umidade atual do solo ou verificar o nível do reservatório, recebendo respostas instantâneas. Além disso, o sistema notificou automaticamente o usuário quando a umidade estava em nível crítico, quando a irrigação foi ativada, e quando o nível de água do reservatório estava baixo. Essas notificações permitiram uma gestão do sistema de irrigação, aumentando a eficiência do processo e a interação do usuário com o sistema.

Figura 27: Exemplos de mensagens Telegram.



Fonte: Elaborado pelo autor, 2024

O chatbot integrado ao sistema de irrigação automatizado foi projetado para fornecer informações claras e úteis ao usuário sobre o status do sistema e seu funcionamento. As mensagens enviadas pelo chatbot têm como objetivo facilitar a interação com o sistema e garantir que o usuário seja informado sobre as condições da planta, do reservatório e do processo de irrigação. Abaixo estão as mensagens que o chatbot pode fornecer:

"Bem-vindo ao irrigador automatizado! Digite um comando:"

Essa mensagem é exibida ao iniciar a interação com o chatbot. Ela oferece instruções para o usuário sobre os comandos disponíveis, como verificar a umidade do solo, o nível do reservatório ou outras funções que podem ser implementadas.

"1. Obter umidade"

Ao selecionar essa opção, o sistema enviará ao usuário o nível atual de umidade do solo, permitindo monitorar as condições da planta e identificar se há necessidade de irrigação.

"2. Obter nível do reservatório"

Essa mensagem fornece informações sobre o nível atual de água no reservatório, garantindo que o usuário saiba quando é necessário reabastecê-lo para evitar o funcionamento a seco da bomba.

"Umidade da planta em nível crítico!"

Essa mensagem é disparada automaticamente quando a umidade do solo está abaixo de 50%, indicando que a planta está em condições de seca e precisa de irrigação urgente.

"Irrigador ligado!"

Informativo automático enviado quando o sistema detecta um nível de umidade inferior ao limite estabelecido (50%) e a bomba é acionada para iniciar a irrigação.

"Irrigador desligado, umidade do solo atingiu o percentual ideal!"

Enviado ao usuário quando a irrigação é interrompida automaticamente após o solo atingir o nível de umidade ideal, confirmando que o ciclo de irrigação foi concluído com sucesso.

"Reservatório em nível crítico, irrigador desligado!"

Essa mensagem alerta o usuário quando o nível de água no reservatório está muito baixo, impedindo o funcionamento do sistema de irrigação e evitando danos à bomba.

"A umidade atual é de X%"

Resposta à solicitação de informações sobre a umidade do solo. O chatbot envia o valor medido pelo sensor em tempo real, facilitando o acompanhamento pelo usuário.

"O nível do reservatório atual é de X%"

Informativo que apresenta o nível de água restante no reservatório em forma de porcentagem, permitindo ao usuário monitorar a necessidade de reabastecimento.

5.2.2 Desempenho do Sensor Ultrassônico

O sensor ultrassônico para medir o nível de água no reservatório proporcionou um monitoramento mais preciso e confiável do volume de água disponível. O sensor foi capaz de medir distâncias entre 2 cm e 4 metros, permitindo uma estimativa percentual do nível de água no reservatório.

Nos testes realizados, o sensor ultrassônico mediu constantemente o nível de água, ajustando a irrigação conforme necessário. Quando o nível de água caiu abaixo de um valor crítico, o sistema enviou um alerta ao usuário via Telegram, evitando assim danos à bomba e garantir a continuidade da irrigação. Este componente se mostrou essencial para a prevenção de falhas no sistema, especialmente em cenários onde o reabastecimento do reservatório não pôde ser feito de maneira imediata.

5.2.3 Indicadores Visuais e Sonoros

Os indicadores visuais (LEDs) e sonoros (buzzer) continuaram desempenhando um papel importante no feedback imediato do sistema, mas com as melhorias, o controle desses elementos foi aprimorado. Agora, além de indicar os níveis de umidade, o sistema também utiliza o buzzer de forma mais eficaz, emitindo diferentes padrões de som para alertar o usuário sobre diferentes condições, como níveis críticos de água ou falhas na comunicação com o sensor.

5.3 Custos de Implementação do Sistema de Irrigação

A implementação do sistema de irrigação automatizado descrito neste trabalho envolveu a aquisição de diversos componentes eletrônicos e hidráulicos, essenciais para o funcionamento do projeto.

Tabela 8: Custo componentes

Componente	Quantidade	Custo Unitário	Custo Total	Fornecedor
ESP32	1	R\$ 64,90	R\$ 64,90	Eletrogate
Sensor de Umidade do Solo	1	R\$ 7,90	R\$ 7,90	Eletrogate
Sensor Ultrassônico HC-SR04	1	R\$ 11,90	R\$ 11,90	Eletrogate
Módulo Relé	1	R\$ 6,90	R\$ 6,90	Eletrogate
Buzzer Ativo	1	R\$ 1,90	R\$ 1,90	Eletrogate
LEDs (Vermelho, Amarelo, Verde)	3	R\$ 0,25	R\$ 0,75	Eletrogate
Resistores (220Ω a 330Ω)	3	R\$ 0,10	R\$ 0,30	Eletrogate
Placas de Acrílico	2	R\$ 19,90	R\$ 39,80	Mercado livre
Protoboard e Jumpers	1	R\$ 14,90	R\$ 14,90	Eletrogate
Fonte de Alimentação 5V	1	R\$ 18,90	R\$ 18,90	Eletrogate
Reservatório de Água	1	R\$ 4,99	R\$ 4,99	Mercado livre
		Total	R\$ 173,14	

Fonte: Elaborado pelo autor, 2024

O custo-benefício deste projeto é altamente favorável, considerando sua eficiência em economizar água, promover a saúde das plantas e simplificar a

manutenção. Além disso, os materiais utilizados são acessíveis e de fácil aquisição, o que reforça a viabilidade do sistema para pequenas aplicações comerciais.

5.4 Possibilidades de outras melhorias

O sistema atual depende de uma fonte de energia elétrica convencional para alimentar o ESP32, os sensores e a bomba d'água. Ao integrar um painel solar fotovoltaico, parte dessa energia pode ser gerada a partir da luz solar, reduzindo o consumo de eletricidade da rede elétrica e tornando o sistema mais sustentável. Com essa melhoria, o sistema poderia operar em locais remotos sem necessidade de energia elétrica constante, reduzindo custos e tornando o projeto mais ambientalmente responsável e economicamente viável.

Figura 28:Mini Painel Solar Fotovoltaico 5V/1W.



Fonte: Eletrogate, 2024

6 CONCLUSÃO

O desenvolvimento deste projeto teve como principal objetivo a criação de um protótipo funcional de um sistema de irrigação automatizado para o cultivo residencial de tomates. A implementação do sistema utilizando Arduino e ESP32 permitiu a automação do processo de irrigação, garantindo que a planta recebesse água apenas dentro da faixa estipulada, de acordo com as condições ideais para seu crescimento saudável.

Durante os testes, o sistema foi capaz de monitorar continuamente os níveis de umidade do solo, atuando dentro da faixa definida entre 50% e 70%, conforme estabelecido nos objetivos do trabalho. O sensor de umidade demonstrou um desempenho preciso, respondendo de forma eficiente às variações do solo e permitindo que o microcontrolador acionasse a bomba de água no momento adequado. Além disso, a inclusão do sensor de nível de água garantiu que a irrigação ocorresse apenas quando houvesse água suficiente no reservatório, protegendo o funcionamento da bomba submersível.

O protótipo operou de maneira satisfatória nos testes, executando corretamente as funções programadas na plataforma Arduino IDE. A utilização de indicadores visuais (LEDs) e sonoros (buzzer) facilitou o acompanhamento do funcionamento do sistema pelo usuário em monitoramento presencial, tornando a interação mais intuitiva. Além disso, a integração com o chatbot no Telegram permitiu o monitoramento remoto, ampliando as possibilidades de controle do sistema.

Dessa forma, o projeto cumpriu seus objetivos, comprovando a viabilidade da automação da irrigação com sensores e microcontroladores acessíveis. O sistema demonstrou ser funcional e adaptável, garantindo a manutenção da umidade do solo dentro da faixa ideal para o desenvolvimento saudável das plantas.

REFERÊNCIAS

ALTOÉ, Marco Antonio. **Sistema automatizado de irrigação para culturas específicas**. 2012. Monografia (Bacharelado em Engenharia de Computação) - Centro Universitário de Brasília (UniCEUB), Brasília-DF, 2012.

ALVARENGA, Marco A. R. Tomate: **Produção em campo, casa de vegetação e hidroponia**. 3. ed. rev. e aum. Lavras, MG: [s. n.], 2022.

ALVES, J. **Como usar a Bomba de Água Diafragma**. Curto-Circuito, 2023. Disponível em: <https://curtocircuito.com.br/blog/eletronica-basica/como-usar-a-bomba-de-agua-diafragma?srsId=AfmBOoqgzmDAHrLpM5pk1IaB0IKWC6CKOacMpgl3LzTX9VjQw2vnB9D9>. Acesso em: 10 fev. 2025.

BARBOSA, José Wilian. **Sistema de irrigação automatizado utilizando plataforma Arduino**. 2013. Monografia (Bacharelado em Ciência da Computação) - Instituto Municipal de Ensino Superior de Assis, Assis, 2013.

DA CUNHA FERREIRA, M. C. **Cultivo de Tomate em Horta Urbana**. Votuporanga: Etec Frei Arnaldo Maria de Itaporanga, 2021.

DA SILVA ROCHA, L. V. **AUTOMAÇÃO RESIDENCIAL: MICROCONTROLADOR ARDUINO ATRAVÉS DE UMA APLICAÇÃO WEB**. BRASÍLIA - DF: CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB, 2019.

DE ARAÚJO, S. I. **STE: Chatbot para auxílio à Secretaria Acadêmica usando Dialogflow e Inteligência Artificial**. Betim - MG: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus Betim, 2022.

LIVIA, C. P. **Como é o Funcionamento das Bombas de Água?** Trans Obra, 2024. Disponível em: <https://www.transobra.com.br/como-e-o-funcionamento-das-bombas-de-agua>. Acesso em: 10 fev. 2025.

MEDEIROS, Pedro. **Sistema de irrigação automatizado para plantas caseiras**. 2018. Monografia (Bacharelado em Engenharia de Computação) - Instituto de

Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2018.

REIS, L. E. P. **DESENVOLVIMENTO DE UM PROTÓTIPO DE CONTROLE DE NÍVEL BASEADO EM ARDUINO: UMA ABORDAGEM UTILIZANDO SENSOR ULTRASSÔNICO**. Patos de Minas - MG: Universidade Federal de Uberlândia, 2024.

SALES, C. X.; DA SILVA, L. M.; AMADEU, S. S. **Sistema de Segurança Inteligente**. São Caetano do Sul - SP: ETEC JORGE STREET, 2017.

SILVA, M. R. V. **PROTÓTIPO DE UM ALIMENTADOR AUTOMATIZADO DE ANIMAIS DE ESTIMAÇÃO MONITORADO E CONTROLADO VIA TELEGRAM**. Betim, Minas Gerais: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus Betim, 2023.

SOARES, F. G. **DESENVOLVIMENTO DE UM PROTÓTIPO DEMONSTRATIVO DE AUTOMAÇÃO RESIDENCIAL, UTILIZANDO MICROCONTROLADOR ESP32**. Campo Grande - MS: Universidade Federal de Mato Grosso do Sul – UFMS, 2023.

SOUZA, M. O. **SENSOR DE NÍVEL TIPO DESLOCADOR COM AUTOCOMPENSAÇÃO DA DENSIDADE DO LÍQUIDO**. São Cristóvão-SE: UNIVERSIDADE FEDERAL DE SERGIPE, 2018.

VIDAL, V. **Automação de Sistema de Irrigação: Sensor de Umidade**. BLOG ELETROGATE, 2017. Disponível em: <https://blog.eletrogate.com/automacao-de-sistema-de-irrigacao-sensor-de-umidade-e-valvula-solenoid/>. Acesso em: 10 fev. 2025.

WILLIAM. **5V Relay Module – How it Works and Application**. GEYA, 2022. Disponível em: <https://www.geya.net/5v-relay-module-how-it-works-and-application>. Acesso em: 10 fev. 2025.

APÊNDICE A

```
// Define o pino analógico que será utilizado para leitura do sensor de umidade do solo.
```

```
#define AnalogPin A0
```

```
// Define os pinos digitais para os LEDs indicativos de níveis de umidade.
```

```
int low = 4; // LED indicando umidade baixa
```

```
int medium = 5; // LED indicando umidade média
```

```
int high = 6; // LED indicando umidade alta
```

```
// Define o pino digital para o controle do relé.
```

```
int Rele = 7; // Pino do relé que controla a bomba de água
```

```
// Define o pino digital para o sensor de nível de água.
```

```
int levelSensor = 8; // Pino do sensor de nível de água
```

```
// Define o pino digital para o buzzer.
```

```
int buzzer = 9; // Pino do buzzer para alarmes sonoros
```

```
// Variável para armazenar a leitura analógica do sensor de umidade do solo.
```

```
int AnalogIn;
```

```
// Variáveis booleanas para controlar o estado do sistema.
```

```
bool on = false; // Indica se o sistema de irrigação está ligado
```

```
bool level = false; // Indica o nível de água
```

```
void setup() {  
  
    // Inicializa a comunicação serial para depuração.  
  
    Serial.begin(9600);  
  
    // Configura os pinos como saídas ou entradas conforme necessário.  
  
    pinMode(Rele, OUTPUT);    // Pino do relé configurado como saída  
  
    pinMode(low, OUTPUT);    // Pino do LED de umidade baixa configurado como  
saída  
  
    pinMode(medium, OUTPUT); // Pino do LED de umidade média configurado  
como saída  
  
    pinMode(high, OUTPUT);    // Pino do LED de umidade alta configurado como  
saída  
  
    pinMode(buzzer, OUTPUT); // Pino do buzzer configurado como saída  
  
    pinMode(levelSensor, INPUT_PULLUP); // Pino do sensor de nível configurado  
como entrada com pull-up interno  
  
    // Inicializa o relé desligado.  
  
    digitalWrite(Rele, HIGH);  
  
    // Atraso para estabilizar o sistema.  
  
    delay(1000);  
  
}  
  
void loop() {
```

```
// Verifica se o nível de água está baixo e se o sistema de irrigação está ligado.
```

```
if (digitalRead(levelSensor) == HIGH && on == true) {
```

```
    // Imprime mensagem de emergência no monitor serial.
```

```
    Serial.println("Emergency Mode, low water level!");
```

```
    // Desliga o relé para parar a bomba de água.
```

```
    digitalWrite(Rele, HIGH);
```

```
    // Ativa o buzzer para emitir um alarme sonoro.
```

```
    tone(buzzer, 440);
```

```
    delay(3000);
```

```
    noTone(buzzer);
```

```
    // Desliga o sistema de irrigação.
```

```
    on = false;
```

```
}
```

```
// Lê o valor do sensor de umidade do solo.
```

```
AnalogIn = analogRead(AnalogPin);
```

```
// Mapeia o valor lido (0-1023) para uma escala de 0 a 100%.
```

```
int value = map(AnalogIn, 1023, 0, 0, 100);
```

```
// Imprime o valor de umidade do solo no monitor serial.
```

```
Serial.print(value);

Serial.println("%");

// Atualiza o estado dos LEDs com base no valor de umidade.

if (value < 50) {

    digitalWrite(low, HIGH); // Acende LED de umidade baixa

    digitalWrite(medium, LOW); // Apaga LED de umidade média

    digitalWrite(high, LOW); // Apaga LED de umidade alta

} else if (value > 50 && value < 60) {

    digitalWrite(low, LOW); // Apaga LED de umidade baixa

    digitalWrite(medium, HIGH); // Acende LED de umidade média

    digitalWrite(high, LOW); // Apaga LED de umidade alta

} else {

    digitalWrite(low, LOW); // Apaga LED de umidade baixa

    digitalWrite(medium, LOW); // Apaga LED de umidade média

    digitalWrite(high, HIGH); // Acende LED de umidade alta

}

// Verifica se a umidade está abaixo do nível ideal e se o sistema de irrigação está
desligado.

if (value <= 50 && on == false) {
```

```
// Verifica se o nível de água está adequado.  
  
if (digitalRead(levelSensor) == LOW) {  
  
    // Imprime mensagem de início de processo no monitor serial.  
  
    Serial.println("Starting the process!");  
  
    // Liga o relé para ativar a bomba de água.  
  
    digitalWrite(Rele, LOW);  
  
    // Define o estado do sistema de irrigação como ligado.  
  
    on = true;  
  
    }  
  
    }  
  
    }
```

APÊNDICE B

```
#include <WiFi.h>
```

```
#include <HTTPClient.h>
```

```
#include <ArduinoJson.h>
```

```
    // SENSOR HUMIDADE - Ok
```

```
    // RELÉ - Ok
```

```
    // PROTEÇÃO - Ok
```

```
    // ULTRASSONICO, 9,86 (min), 2,32 (max) - Ok
```

```
    // LEDS + BUZZER - Ok
```

```
String myId = ""; //HERE - trocar pelo ID do seu chat telegram
```

```
bool sent = false;
```

```
bool on = false;
```

```
bool isCriticalsent = false;
```

```
//pinout
```

```
const int sensorPin = 34; // Pino onde o sensor de umidade está conectado
```

```
const int low = 25; // led vermelho
```

```
const int medium = 26; // led amarelo
```

```
const int high = 27; // led verde
```

```
const int rele = 14;

const int levelSensor = 12; //sensor de proteção da bomba d'água

const int buzzer = 13;

const int frequency = 440; // Frequência do tom (Hz)

const int ledcChannel = 0; // Canal PWM

const int ledcTimer = 0; // Timer PWM

const int trigPin = 18; // Pino TRIG do HC-SR04

const int echoPin = 19; // Pino ECHO do HC-SR04

//HERE - Substitua com suas credenciais de rede Wi-Fi

const char* ssid = "";

const char* password = "";

//HERE - Substitua pelo token do seu bot

const char* botToken = "";

const unsigned long requestInterval = 10000; // 10 segundos

unsigned long lastRequestTime = 0;

int lastUpdateId = 0; // ID da última atualização processada

void setup() {

    Serial.begin(115200);
```

```
pinMode(low, OUTPUT);

pinMode(medium, OUTPUT);

pinMode(high, OUTPUT);

pinMode(rele, OUTPUT);

pinMode(buzzer, OUTPUT);

ledcSetup(ledcChannel, frequency, 8); // Canal, Frequência (Hz), Resolução (8 bits)

ledcAttachPin(buzzer, ledcChannel); // Anexa o pino ao canal PWM

pinMode(levelSensor, INPUT_PULLUP);

digitalWrite(rele, HIGH);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

    // Conectar-se à rede Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}

Serial.println("Conectado ao Wi-Fi!");
```

```
}  
  
void loop() {  
  
    getUpdates();  
  
    if(digitalRead(levelSensor) == HIGH && on == true)  
  
    {  
  
        Serial.println("Emergency Mode, low water level!");  
  
        digitalWrite(rele, HIGH);  
  
        ledcWriteTone(ledcChannel, frequency);  
  
        delay(3000);  
  
        ledcWriteTone(ledcChannel, 0);  
  
        on = false;  
  
    }  
  
    int actual = getSoloStatus();  
  
    Serial.println(actual);  
  
    if(actual < 50){  
  
        if(!sent){  
  
            sendTelegramMessage("Umidade da planta em nível crítico!", myId);  
  
            sent = true;  
  
        }  
  
    }  
  
}
```

```
}  
  
    digitalWrite(low, HIGH);  
  
    digitalWrite(medium, LOW);  
  
    digitalWrite(high, LOW);  
  
}else if(actual > 50 && actual < 60){  
  
    sent = false;  
  
    digitalWrite(low, LOW);  
  
    digitalWrite(medium, HIGH);  
  
    digitalWrite(high, LOW);  
  
}else{  
  
    sent = false;  
  
    digitalWrite(low, LOW);  
  
    digitalWrite(medium, LOW);  
  
    digitalWrite(high, HIGH);  
  
}  
  
if(actual <= 50 && on == false){  
  
    if(digitalRead(levelSensor) == LOW)  
  
    {
```

```
digitalWrite(rele, LOW);

sendTelegramMessage("Irrigador ligado!", myId);

on = true;

isCriticalsent = false;

}else

{

Serial.println("Starting the alarm!");

if(!isCriticalsent)

{

    sendTelegramMessage("Reservatório em nível critico, irrigador desligado!",
myId);

    isCriticalsent = true;

}

    for(int i = 0; i < 10; i++)

{

digitalWrite(low, HIGH);

delay(150);

digitalWrite(low, LOW);

delay(150);
```

```

        Serial.println("alarme running");

    }

}

}else if (actual > 45 && on == true){

    digitalWrite(rele, HIGH);

    sendTelegramMessage("Irrigador desligado, umidade do solo atingiu o
percentual ideal!", myId);

    on = false;

    isCriticalsent = false;

}

delay(2000);

}

void getUpdates() {

    if (WiFi.status() == WL_CONNECTED) {

        HTTPClient http;

        String url = "https://api.telegram.org/bot" + String(botToken) + "/getUpdates";

        if (lastUpdateId > 0) {

            url += "?offset=" + String(lastUpdateId + 1); // Solicita apenas novas mensagens

        }

    }

}

```

```
    http.begin(url);

int httpCode = http.GET();

if (httpCode > 0) {

    String payload = http.getString();

    // Serial.println("Resposta do Telegram:");

    // Serial.println(payload);

    DynamicJsonDocument doc(1024);

    DeserializationError error = deserializeJson(doc, payload);

    if (!error) {

        JSONArray messages = doc["result"].as<JSONArray>();

        for (JsonObject message : messages) {

            int updateId = message["update_id"];

            String text = message["message"]["text"].as<String>();

            String chatId = message["message"]["chat"]["id"].as<String>();

            JsonObject from = message["message"]["from"];

            String firstName = from["first_name"].as<String>();

            Serial.println(text);

            Serial.println(firstName);
```

```
        Serial.println(chatId);

        lastUpdateId = updateId;

        defineMessageType(text, firstName, chatId);

    }

} else {

    Serial.println("Falha ao analisar JSON.");

}

} else {

    Serial.println("Erro ao obter atualizações.");

}

http.end();

} else {

    Serial.println("Erro de conexão Wi-Fi.");

}

}

void sendTelegramMessage(String message, String chatId) {

    if (WiFi.status() == WL_CONNECTED && chatId != "") {

        HTTPClient http;
```

```
String url = "https://api.telegram.org/bot" + String(botToken) +
"/sendMessage";

http.begin(url);

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

String payload = "chat_id=" + chatId + "&text=" + message;

int httpResponseCode = http.POST(payload);

if (httpResponseCode > 0) {

String response = http.getString();

Serial.println("Mensagem enviada com sucesso!");

Serial.println(response);

} else {

Serial.println("Erro ao enviar mensagem.");

Serial.println(httpResponseCode);

}

http.end();

} else {

Serial.println("Não conectado ao WiFi ou chat ID não definido.");

}

}
```

```
void defineMessageType(String comando, String nome, String chatId) {

    Serial.println(comando);

    if (comando == "1") {

        Serial.println("obter humidade");

        int actual = getSoloStatus();

        String message = "Olá " + nome + ", a umidade atual é de " + String(actual) + "%";

        sendTelegramMessage(message, chatId);

    }else if( comando == "2"){

        Serial.println("obter nivel");

        float levelPercentage = getLevelInPercentage();

        String message = "Olá " + nome + ", o nivel do reservatorio atual é de " +
String(levelPercentage) + "%.";

        sendTelegramMessage(message, chatId);

    }

    // else if( comando == "3"){

    //   Serial.println("irrigar");

    //   String message = "Olá " + nome + ", o irrigador foi ligado.";

    //   sendTelegramMessage(message, chatId);

    // }
```

```
// else if( comando == "4"){

// Serial.println("irrigar");

// String message = "Olá " + nome + ", o irrigador foi desligado.";

// sendTelegramMessage(message, chatId);

// }

else{

    Serial.println("boas vindas");

    String message = "Olá " + nome + ", Bem-vindo ao irrigador automatizado! Digite
um comando:\n\n"

        "1. Obter umidade\n"

        "2. Obter nível do reservatório\n";

    sendTelegramMessage(message, chatId);

}

}

int getSoloStatus(){

int AnalogIn = analogRead(sensorPin);

int value = map(AnalogIn, 0, 4095, 0, 100);

return value;

// Serial.print(value);
```

```
// Serial.println("%");  
  
}  
  
float getLevelInPercentage() {  
  
    float level = getTankLevel();  
  
    const float minLevel = 2.3; // 100% em cm  
  
    const float maxLevel = 9.86; // 0% em cm  
  
    if (level < minLevel) level = minLevel;  
  
    if (level > maxLevel) level = maxLevel;  
  
    float percentage = (maxLevel - level) / (maxLevel - minLevel) * 100.0;  
  
    return percentage;  
  
}  
  
float getTankLevel()  
  
{  
  
    digitalWrite(trigPin, LOW);  
  
    delayMicroseconds(2);  
  
    digitalWrite(trigPin, HIGH);  
  
    delayMicroseconds(10);  
  
    digitalWrite(trigPin, LOW);
```

```
long duration = pulseIn(echoPin, HIGH);  
  
float distance = (duration / 2.0) * 0.0343;  
  
Serial.print("tank level: ");  
  
Serial.print(distance);  
  
delay(500);  
  
return distance;  
  
}
```